PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 17/30

A1

(11) International Publication Number: WO 99/34307

(43) International Publication Date: 8 July 1999 (08.07.99)

(21) International Application Number: PCT/US98/27664

(22) International Filing Date: 28 December 1998 (28.12.98)

(30) Priority Data:
60/068,920 29 December 1997 (29.12.97) US

(71) Applicant *(for all designated States except US)*: INFODREAM CORPORATION [US/US]; 2340A Walsh Avenue, Santa Clara, CA 95051 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: ANDLEIGH, Prabhat, K. [US/US]; 10701 Castine Avenue, Cupertino, CA 95014 (US). PAPPU, Nagaraju [IN/US]; Apartment 14 H, 20800 Homestead Road, Cupertino, CA 95014 (US). KALIDINDI, Vasudeva, V. [IN/US]; Apartment 95, 3655 Pruneridge Avenue, Santa Clara, CA 95051 (US).

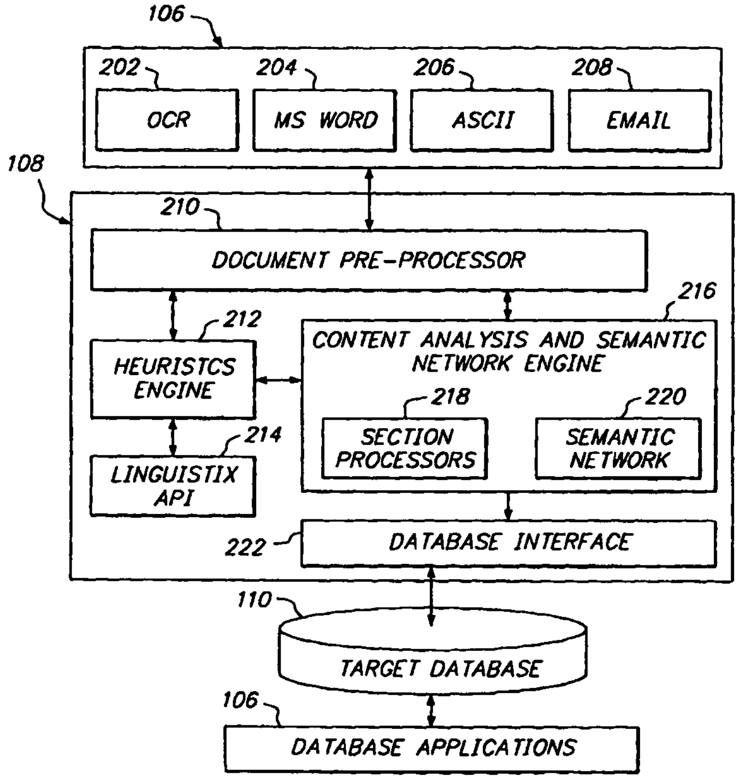
(74) Agents: ARRIOLA-KERN, Trinidad et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: EXTRACTION SERVER FOR UNSTRUCTURED DOCUMENTS

(57) Abstract



A system for analyzing and extracting words and word groups from an electronic document (104) and for storing the extracted words and word groups into predefined fields or tables in a target database (110) comprises a content analysis and semantic network engine (216) for analyzing and extracting words and word groups from the electronic document and a heuristics engine (212) coupled to the content analysis and semantic network engine (216), for applying a set of heuristics to the words and word groups in the electronic document. The content analysis and semantic network engine (216) further comprises a thesaurus (400) for linking together terms (402) and concepts (404) and for defining relationships between and among the terms (402) and concepts (404), a semantic network (220) coupled to the thesaurus (400), for organizing the terms (402) and concepts (404) in the thesaurus (400), meta-concepts (502), and categories (504) in a hierarchical structure, and section processors (218) for analyzing a section in the electronic document (104) and applying a set of heuristics to each section in the electronic document (104). The system further comprises a document pre-processor (210) for performing an initial analysis on the electronic document (104), a morphological analysis engine (214) coupled to the heuristics engine (212) for performing a morphological analysis and tagging of words and word groups in the electronic document (104), and a database interface (222) for providing an interface between the content analysis and semantic network engine (216) and the target database (110).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

EXTRACTION SERVER FOR UNSTRUCTURED DOCUMENTS

Related Application

The subject matter of this application is related to and claims priority from U.S. Provisional Application Serial No. 60/068920 entitled "Auto Entry Server" by Prabhat K. Andleigh, Nagaraju Pappu and Vasudeva Kalindindi, which was filed on December 29, 1997 and which subject matter is incorporated herein by reference in its entirety.

Technical Field

This invention relates to the field of computer analysis of electronic documents. More specifically it relates to the field of information retrieval to convert and store information in documents written in a natural language into a predefined structure which can be retrieved and manipulated by computer program applications.

Background of the Invention

Information to be sorted and stored in a computer database may reside in numerous electronic documents. For example, information about people and their specific talents and skills may reside in electronic documents, such as resumes, performance appraisals, design documents, publications, books, patent documents, and email messages. When an individual is trying to organize and sort out specific information from such electronic documents, the individual usually has to open each document separately and manually analyze, retrieve, and store the relevant data in the particular database. For example, a project manager who would like to find the best employee for a specific job may have a specific job description. When searching for an employee whose skills, knowledge and talent are best suited for the specific job description, the project manager must sift through several documents which contain the necessary information. Such a process is time consuming and inefficient because the project manager may have to read the documents several times and may have to review and type the information into a computer database in order to organize the various pieces of information into a coherent summary.

A computerized system which can analyze and extract pertinent information from different electronic documents would provide a more efficient solution to this problem. However, such text documents are often written in unstructured natural language text for other people to understand. Thus, computer programs such as database applications cannot efficiently process documents written in natural language texts. Rather, computer programs can process only information which has been stored in a highly structured fashion in order to retrieve and manipulate that information. Additionally, these documents may be prepared in a variety of different file formats, such as Microsoft Word 97, Rich Text Format, PDF, WordPerfect, ASCII files, and HTML, and may be stored in different areas within a computer.

There are a variety of information retrieval programs such as Internet search engines that can retrieve documents that match a set of keywords. Their scope is very limited in the context of the above mentioned problem. because they cannot understand the text, and certainly they cannot make any connection between the document and the person who is related to that document. Another problem is that the 'information of interest' will vary significantly from one organization to another. For example, a health care organization will be interested in the skills and talents related to the medical field, but the skills related to computers may not be of significant interest, whereas a software development organization will be interested in the computer and software related skills. but may not be interested in medical or first-aid related skills. The keyword based search engines cannot address this problem of retrieving only the 'information of interest'. As a result. there is a vast amount of information about people which cannot be easily processed by computer programs.

Therefore, what is needed is a system for analyzing and extracting information from an electronic document and for storing the extracted information in a database. Additionally, what is needed is a system for analyzing and extracting information from an electronic document which can process an electronic document irrespective of the original file format, which is language independent. and which can process any type of document.

Disclosure of Invention

The present invention is a system and method for analyzing and extracting words and word groups from an electronic document (104) and for storing the extracted words and word groups into predefined fields or tables in a target database (110). The system for analyzing and extracting words and word groups from an electronic document (104) comprises a content analysis and semantic network engine (216) for analyzing and extracting words and word groups from the electronic document and a heuristics engine (212) coupled to the content analysis and semantic network engine (216), for applying a set of heuristics to the words and word groups in the electronic document. The content analysis and semantic network engine (216) further comprises a thesaurus (400) for linking together terms (402) and concepts (404) and for defining relationships between and among the terms (402) and concepts (404), a semantic network (220) coupled to the thesaurus (400), for organizing the terms (402) and concepts (404) in the thesaurus (400), meta-concepts (502), and categories (504) in a hierarchical structure, and section processors (218) for analyzing a section in the electronic document (104) and applying a set of heuristics to each section in the electronic document (104).

The system further comprises a document pre-processor (210) for performing an initial analysis on the electronic document (104), a morphological analysis engine (214) coupled to the heuristics engine (212) for performing a morphological analysis and tagging of words and word groups in the electronic document (104), and a database interface (222) for providing an interface between the content analysis and semantic network engine (216) and the target database (110).

A method for extracting words and word groups from an electronic document comprises the steps of: identifying a section in the electronic document (602); analyzing and extracting words and word groups in the section in the electronic document (604); and storing words and word groups extracted from the section into a target database (606). The method further comprises the steps of: converting the electronic document from a native file format into an ASCII text format (302); filtering out unnecessary information from the electronic document (304); and identifying sections of interest in the electronic document (310).

Brief Description of the Drawings

Figure 1 is a block diagram of a preferred embodiment of a system in accordance with the present invention.

Figure 2 is a block diagram of a preferred embodiment of an xtraction server in accordance with the present invention.

Figure 3 is a flow chart of a preferred embodiment of the steps performed by the document pre-processor.

Figure 4 is a block diagram of a preferred embodiment of a thesaurus.

Figure 5 is a block diagram of a preferred embodiment of a semantic network.

Figure 6 is a flow chart of a preferred embodiment of the steps performed by the xtraction server.

Figure 7 is a block diagram of a preferred embodiment of the section processors for a resume document type.

Detailed Description of the Preferred Embodiments

Referring now to Figure 1, a system upon which a preferred embodiment of the present invention operates is shown. A host computer 102, using the method and system described herein, operates upon an electronic document 104, derived from a text document which contains unstructured text. As used herein "unstructured text" refers to any document which has been written in a natural language such as English. Examples of documents containing unstructured text include, but are not limited to, a resume, performance appraisals, design documents, publications, books, patent documents, and email messages. In a preferred embodiment, the host computer 102 is a conventional computer having a keyboard and mouse for input, and a conventional memory 106 associated with host computer 102 for storing the electronic document 104. The electronic document 104 may be prepared in any electronic file format, such as Microsoft Word 97, Rich Text Format, PDF, WordPerfect, ASCII files, and HTML.

The electronic document 104 is processed by host computer 102 using the present invention. Specifically, host computer 102 uses xtraction server 108 to analyze, retrieve and store words and word groups from the electronic document 104 into a predefined structure in target database 110. As used herein, the terms "words" and "word groups" are used to mean any text that may be derived from document 104 including, but not limited to, individual words or numbers, phrases, whole sentences, and blocks of text. The xtraction server 108 identifies the document type of the document 104 and determines which words and word groups are to be extracted from the document 104. The structure and operation of the extraction server 108 is described in more detail below with reference to Figures 2 through 6.

The target database 110 comprises predefined tables with predefined columns for storing the word and word groups extracted from the electronic document 104. In a preferred embodiment, a predefined table and predefined columns correspond to a particular document type. For example, if document 104 is a resume, then a predefined table for a document type called "resumes" may have predefined columns such as "name", "address", "education", and "experience". As another example, if document 104 is a patent document. then a predefined table for a document type called "patent document" may have predefined columns such as "inventors", "company", "patent number", and "field of search". The predefined tables and columns in target database 110 are organized ahead of time, and one skilled in the art will realize that the present invention is not limited to a particular document type or a predefined table but that many different compilations of predefined tables and columns may be stored in target database 110 within the scope of this invention. The words and word groups stored in the target database 110 can be stored in electronic form on any type of computer data storage device or they may printed out in a hard-copy printed format. The target database 110 is described below in more detail with reference to Tables 9 through 15.

The process of extraction performed by the xtraction server 108 uses a non-monotonic reasoning principle. As used herein. a "non-monotonic reasoning principle" refers to a process whereby at every stage during extraction, the xtraction server 108 assumes a reasonable default value. That default value is modified as further information becomes available. For example, a string '1987' is first assumed to be a number, and if further information to qualify the string to be a date is available (for example in this case, that the string is preceded by another string 'Jan'), then the assumption is changed. If again further information becomes available to negate the previous assumption. the assumption is changed again.

Thus, the present invention advantageously allows a user to extract information from an electronic document directly into a database. More specifically, the present invention analyzes an electronic copy of a text document and extracts words and word groups into a target database comprising predefined tables and columns associated with a particular document type. Moreover, the present invention operates upon electronic documents in any electronic file format. The extracted information stored in the target database can then be retrieved and manipulated by other computer program applications.

Referring now to Figure 2. a block diagram of a preferred embodiment of the xtraction server 108 is shown. The electronic document 104 may be any electronic file stored in memory 106 which is accessible by the xtraction server 108. For example. the electronic document 104 may be an electronic form of a hard copy of a document converted using a conventional optical scanner and Optical Character Recognition (OCR) software 202, a Microsoft Word file 204, an ASCII text file 206 or an email attachment 208. The database applications which manipulate the extracted information in target database 110 is also preferably stored in memory 106. In a preferred embodiment, the xtraction server 108 comprises a document preprocessor 210 coupled to the memory 106 where the electronic document 104 is stored, a heuristics engine 212 coupled to the document pre-processor 210, a morphological analysis engine 214 coupled to the heuristics engine 212, a content analysis and semantic network engine 216 coupled to the document preprocessor 210, and a database interface 222 coupled to the content analysis and semantic network engine 216 and to the target database 110. The content analysis and semantic network engine 216 preferably comprises section processors 218 and a semantic network 220.

The document pre-processor 210 retrieves the electronic document 104 from memory 106 and performs the initial analysis of the electronic document 104. Referring now to Figure 3, a flowchart of the steps of a preferred operation of the document pre-processor is shown. The document pre-processor 210 performs the initial analysis and extraction of the electronic document 104 by first converting (302) the electronic document 104 from its native file format into ASCII text. More specifically, the document pre-processor identifies the file format of the electronic document 104 and extracts the ASCII text out the document 104. For example, if the electronic document 104 is a Microsoft Word file, then the document pre-processor 210 identifies the file by the Microsoft Word signature and uses the Microsoft Object Linking and Embedding Software Development Kit (Microsoft OLE 2.0 SDK) to extract text from the Microsoft Word File.

Next, the document pre-processor 210 filters out (304) any unnecessary and unwanted information such as, but not limited to, email headers. OCR headers, blank pages, and unwanted characters. Preferably, any information that is not part of the original document is treated as unnecessary information. For example, email headers, non-ASCII characters at the beginning or at the end of the file, extra blank lines and blank spaces are removed from the text. Additionally, if the text contains vertical tables, these tables are preferably converted into horizontal tables. If the text contains multiple columns, it is preferably converted into single column. The document pre-processor 210 then stores (306) formatting information for the document 104 such as, but not limited to, the fonts used, font sizes, section tittles, and subsections.

The document pre-processor 210 then performs paragraph identification heuristics (308) on the electronic document 104. During this step, the beginning and end of each paragraph is identified, and the paragraph characteristics are gathered. As used herein, the phrase "paragraph characteristics" refers to the statistical properties of the paragraph. Paragraph characteristics include, but are not limited to, the number of words in the paragraph, the number of lines in the paragraph, the average number of words per line, whether any line has a bullet as the starting character, and whether there are any underlined sentences in the paragraph.

Finally, the document pre-processor 210 performs paragraph grouping heuristics (310) on the electronic document 104. Once the paragraphs have been identified, the document pre-processor 210 groups the paragraphs into sections. During this step, the paragraphs are grouped into sections based on the paragraph characteristics as well as using any section tittles that precede the paragraphs. Starting at the beginning of the electronic document 104, the first heading or section tittle is identified, and the following paragraphs until the next section title are grouped into one section. If no section titles are found, then using the paragraph characteristics, all the similar paragraphs are grouped into sections. Additionally, paragraphs that have same or similar characteristics are grouped together into sections.

The heuristic engine 212 applies a set of heuristics, that is a set of rules, to the electronic document 104 for analyzing information in the electronic document 104. The set of heuristics which are applied to the electronic document 104 are associated with a particular document type. For example, if the document type is a "resume", then the set of heuristics associated with the document type "resume" is applied to the electronic document 104. Heuristics are described below in more detail with reference to the section processors.

The morphological analysis engine 214 is used for target language analysis and is preferably the LinguistiX 2.0 application programming interface (API) from InXight Corporation in PaloAlto, CA. The LinguistiX 2.0 API is a language neutral programming interface. In other words, the LinguistiX API can analyze documents in any language such as English, French or German. Because the heuristics engine 212 and the LinguistiX API are external to and separate from the document pre-processor 210 and the content analysis and semantic network engine 216, the present invention can extract information from documents in the English, French or German language, and any other languages which will be supported by the LinguistiX API in future.

Preferably, the Heuristics Engine 212 uses the following features provided by the LinguistiX API: tokenization, lexical analysis, tagging, and noun-phrase extraction. Before text from the electronic document 104 can be analyzed in terms of its linguistic roots and function, it must first be segmented into words, punctuation and idiomatic phrases. LinguistiX tokenization includes the ability to recognize multi-word constructs such as HTML tags. The lexical analysis feature identifies the grammatical features of a word in addition to its root forms. The tagging feature identifies the grammatical category of words by their context. The noun-phrase extraction identifies multi-word phrases in documents. LinguistiX phrase extraction technology enables software to work with these larger concepts to provide improved information analysis and retrieval. For example, 'Windows Programming' will be identified as one phrase, instead of two distinct words Windows and Programming. This feature is used by the semantic network 220 to identify the multi-word noun phrases.

These features of the LinguistiX API are used to implement the heuristics that are described in later sections. For example, by using the tagging feature, the xtraction server 108 may discover that a particular word is a proper noun. Whether that word is the name of the person or the name of a company will depend on where the word occurred in a document. For example, if the word occurs in a contact information section of a document, then it may be the name of the person, or name of the street, city and so on. If the word occurs in an experience section of a document, and if it is followed by the name of a city and state, it may be a company name.

The database interface 222 is a set of APIs that provide a mechanism for retrieving and storing information to and from the target database 110. This is done in such a way that the underlying implementation of the target database 110 is hidden from the application using the database interface. Thus, the xtraction server 108 can work with any industry standard relational database software such as Oracle or Microsoft SQL Server without having to change the software or its implementation. Additionally, the database interface 222 provides the following mechanisms: a method to connect to the target database, a method to maintain the connection to the database, a transaction model to maintain the consistency of the database, and various methods to retrieve, query, update, insert and delete information from the target database 110.

The content analyzer and semantic network engine 216 analyzes the content of the electronic document 104, extracts words and word groups from the document 104, and stores the extracted information in the appropriate tables in the target database 110. In a preferred embodiment, the content analyzer and semantic network engine 216 comprises section processors 218 which extract information from a particular section of interest, and a semantic network 220. The semantic network 220 uses a thesaurus (not shown) and a phrase extraction process to identify the meta-concepts and categories in the electronic document 104 and extracts related words and word groups into the target database 110.

Thesaurus

The thesaurus is a vocabulary database for the extraction server 108 and is organized by concepts. Referring now to Figure 4, a block diagram of a preferred embodiment of a thesaurus 400 is shown. The thesaurus 400 groups all related terms 402 in a language under a language independent concept 404. As used herein, a "term" 402 refers to all the individual words or word groups that belong to a particular language along with their alternatives. As used herein, a "concept" 404 comprises of a set of terms 402 that are language specific and alternatives to one another. However, the concept 404 itself is language independent. Concepts 404 establish synonymous relationships among all terms 402 in the thesaurus 400 that have the same meaning. In other words, concepts 404 connect all the different names for the same concept 404 that are known to the thesaurus 400 and specify certain characteristics for each name. Preferably, each concept 404 has a unique concept identifier (ConceptID). The Concept ID by itself has no intrinsic meaning. Each term 402 in each language in the thesaurus 400 has a unique term identifier. The same term 402 in different languages, for example, in English and Spanish, will have a different term identifier for each language.

To illustrate the relation between terms 402 and concepts 404 consider an example in which term1 402A may consist of 'MS VC++', term2 402B may consist of 'Microsoft Visual C++' and term3 402C may consist of 'MS Visual C++'. All these terms 402 are linked to the concept 404 'Visual C++'. In other words, if the electronic document 104 uses any of the words or word groups 'MS VC++', 'Microsoft Visual C++' or 'MS Visual C++', the thesaurus 400 allows the xtraction server 108 to recognize the words or word groups as being linked to the concept1 404A 'Visual C++'. In another example, term4, term5 and term6 are respectively 'JDK 1.1', 'Symantec Café', and 'JDBC', and all these terms 402 are linked to the concept2 404B called 'Java'. Thus, if the electronic document 104 uses any of the words or word groups 'JDK 1.1', 'Symantec Café', and 'JDBC', the thesaurus 400 allows the xtraction server 108 to recognize the word or word group as being linked to the concept2 404B 'Java'.

The thesaurus 400 may also comprises other information such as the attributes of a concept 404 or attributes of a term 402. Attributes provide additional information that helps to define the meaning of a concept 404 and explain how it may be used in a document. In other words, the different senses of a particular word or word groups is captured using the attributes.

In addition to the relationship between a concept 404 and a set of terms 402, the thesaurus 400 also comprises relationships among concepts 404. Preferably, these relationships are non-subsumption relationships. As used herein, the term "non-subsumption" refers to relationships that include related concepts, co-occurring concepts and/or associated expressions. In other words, non-subsumption refers to relationships that are not based on subsumption. For example, C++ and Java are related, but neither subsumes the other. All these relationships among concepts 404 indicate that the concepts 404 linked together are not exactly similar but are associated with each other in different ways. One skilled in the art will realize that the terms and concepts of the thesaurus 400 are not limited to the examples given herein but may contain any number of terms and concepts which have been predefined and stored in the thesaurus 400 prior to the processing of the electronic document 104. Thus, the thesaurus advantageously allows the present invention to link together terms and concepts used in specific industries, disciplines, and technologies for which the thesaurus is being used and preserves the meanings and hierarchical connections between those terms and concepts. Additionally, the thesaurus facilitates the access to concept relationships and to term and concept attributes irrespective of the term used as a point of entry.

Semantic Network

The semantic network 220 provides a way of arranging all the concepts 404 at the lowest level and then builds a taxonomy or network of higher level meta-concepts and categories. Referring now to Figure 5, a block diagram of a preferred embodiment of a semantic network 220 is shown. The semantic network 220 comprises concepts 404 at the lowest level, meta-concepts 502 at a second level, and categories 504 at the highest level. The semantic network 220 together with the thesaurus 400 provides a four level hierarchy of terms 402, concepts 404, meta-concepts 502 and categories 504.

A category 504 is the highest level in the semantic network 222. Broad categories 504 may be created according to a specific industry which fully subsume other meta-concepts 502 and concepts 404. The semantic network 220 categorizes all meta-concepts 502 into categories 504. Meta-concepts 502 comprise the next level in the semantic network 220 hierarchy. Each meta-concept 502 is a collection of concepts 404 that add to the body of knowledge. The semantic network 220 categorizes all concepts 404 into meta-concepts 502. As described earlier with reference to Figure 4, concepts 404 are generic and language independent from all related terms 402. The semantic network 220 categorizes all terms 402 into concepts 404. As described earlier with reference to Figure 4, terms 402 comprise language dependent strings that are found in the electronic document 104. Terms 402 comprise the lowest level in the semantic network 220 hierarchy.

The entire semantic network 220, separate from the thesaurus 400, comprises language independent knowledge that is arranged as a taxonomy. Preferably, the relationships between concepts 404 and meta-concepts 502 as well as the relationships between meta-concepts 502 and categories 504 are many to many. In other words, a single meta-concept 502 can comprise several concepts 404 and a single concept 404 can be linked to several meta-concepts 502. Similarly, several meta-concepts 502 may comprises a category 504 and several categories may have links to a single meta-concept 502.

To illustrate the terms 402, concepts 404, meta-concepts 502, and categories 504 of a semantic network 220, the two concepts discussed earlier with reference to Figure 4, namely 'Visual C++' and 'Java', will be used. Both these concepts 404 may be grouped under a meta-concept 502 'Object Oriented programming languages'. Additionally, the concept 404 'Visual C++' may also belong to the meta-concept 502 'Visual Programming Environment'. The meta-concept 502 “Visual Programming Environment” may also be linked to other concepts 404 such as 'Visual Basic'.

The semantic network 220 uses subsumption as the basis for the hierarchical organization of concepts 404, meta-concepts 502, and categories 504. In other words, the relationship between concepts 404 and meta-concepts 502 and meta-concepts 502 and categories 504 in the semantic network 220 are based on conceptual subsumption, where a more general object 'subsumes' a more specific object. The concept of subsumption is more general than the concept of synonymy. An object is subsumed by another object if the subsuming object is much more general than any other subsumed objects and effectively summarizes the subsumed objects. Truly synonymous objects mutually subsume each other. If only synonymous based relationships are allowed then the granularity between the objects cannot be captured effectively as there are not many truly synonymous objects. The difference between the shades of meaning will not allow correct retrieval in a synonym-based network. The subsumption-based network removes these drawbacks and aids in retrieving related concepts more accurately, since a subsumption is more general compared to a synonym. For example, the object 'JDBC' is subsumed by a more general object called 'Java Programming Language' (a meta-concept 502), which is further subsumed by an even more generic object 'Software Engineering' (a category 504).

An object may also be subsumed by more than one higher level object. For example, the concept 404 'JDBC' may be subsumed by at least two meta-concepts 502 such as 'Java Programming Language' and 'Database Connectivity Library'. Each of these meta-concepts 502 may in turn be subsumed by several categories 504. Hence, the conceptual subsumption also allows many-to-many relationships between concepts 404 and meta-concepts 502 and between meta-concepts 502 and categories 504.

Referring now to Figure 6, a flowchart of the steps of a preferred embodiment of a method performed by the content analysis and semantic network engine 216 is shown. First, identification heuristics are performed (602) on the electronic document 104 to identify the beginning and end of the known sections of interest. The sections of interest are configured by the user when the xtraction server 108 is first installed. The sections are then analyzed (604) and information is extracted from the sections. The extracted information is stored (606) in a predefined structure in the target database 110. Using the semantic network 220, words and word groups are analyzed (608) and the relationships between the different words and word groups are determined and stored in the target database 110. Thus, the present invention advantageously extracts meaningful information from electronic documents, and stores them in a predefined structure in a target database. The extracted information stored in the target database can then be retrieved and manipulated by computer program applications accessing the database. Moreover, the present invention provides a powerful semantic network and thesaurus for defining terms, concepts, meta-concepts, and categories and the relationship between and among such terms, concepts, meta-concepts, and categories. Thus, the semantic network can stored information relating to any field, industry or technology, and allows the xtraction server 108 to process various types of documents pertaining to such fields, industries or technologies.

Database Tables for the Thesaurus and Semantic Network

In a preferred embodiment, database tables are used to define how the semantic network 220 and the thesaurus 400 are represented in a relational or object oriented database. In an object-oriented implementation, any relational table is preferably represented as an object class. The following tables define several aspects of the semantic network 220 and the thesaurus 400. One skilled in the art will realize that these tables are not limited to the specific information illustrated therein but may be created. as needed depending on the document type being processed. Tables 1, 2 and 3 provide preferred terms and definitions for the thesaurus 400. More specifically, Table 1 defines the terms 402 which are language specific.

Table 1

TermID	Unique identifier of the term
TETerm	The string representation of the term
LanguageID	Unique identifier of the language to which the term belongs
ConceptID	Unique identifier of the concept to which the term is subsumed

Table 2 stores information about different languages to which the terms 402 belong.

Table 2

LanguageID	Unique identifier of the language
LALanguage	Name of the language

Table 3 stores information relating to concepts 404.

Table 3

ConceptID	Unique identifier of the concept
CNConceptName	Name of the concept
CNDescription	Description of the concept
CNSemMarer	Semantic markers of the concept

Tables 4 through 7 provide preferred terms and definitions for the semantic network 220.

Table 4

MetaConceptID	Unique identifier of the concept
MEMetaConceptName	Name of the meta-concept
ME Description	Description of the meta-concept
MESemMarer	Semantic markers of the meta-concept

Table 5 represents the relationships between the concepts 404 and meta-concepts 502.

Table 5

MetaConceptID	Unique identifier of the meta-concept
ConceptID	Unique identifier of the concept
CRRelationType	The type of the relationship between the concept and meta-concept
CrisaRealtionYN	Yes or no value whether the concept is subsumed by meta-concept
CRDescription	Description of the relation

Table 6 provides information relating to the categories 504 in the semantic network 220.

Table 6

CategoryID	Unique identifier of the category
CTcategoryName	Name of the category
CTDescription	Description of the category
CTSemMarer	Semantic marker of the category

Table 7 provides information regarding the relationships between categories 504 and meta-concepts 502.

Table 7

MetaConceptID	Unique Identifier of the meta-concept
CatgeoryID	Unique identifier of the category

Table 8 provides a list of defined values for the semantic markers, that is, attributes of the concepts 404, meta-concepts 502. and categories 504.

Table 8

LT	Lexical Tag
SCT	Syntactic category/parts of speech
ST	Concept attribute status
ANIM	Animate object
INANIM	Inanimate object
PHYS	Physical object
ABST	Abstract object

Thus, using Tables 1 through 8, the content analysis and semantic network engine 216 identifies terms 402 in the electronic document 104, identifies concepts 404 and the relationships between the concepts 404 and the terms 402. identifies the lowest concept 404 and meta-concept 502 that subsumes the given concepts, extracts all related meta-concepts 502 and categories 504, and generates the values for the SQL statements to update the target database 110.

Section Processors

The section processors 218 extract information from sections of interest in an electronic document 104. The particular sections of interest from which information is extracted is determined by the document type. The content analysis and semantic network engine 216 comprises a section processor 218 for extracting words or word groups from each section of interest in an electronic document.

To illustrate the functionality of the section processors 218, the operation of the section processor 218 on a specific document type called "resume" is used in the following sections. Resumes typically contain several sections such as a cover letter, contact information, an objective section, an experience section, an education section, a patents section, a publications section, an awards and honors received section, and a courses attended section. Referring now to Figure 7, a block diagram of a preferred embodiment of section processor 218 for a resume document type is shown. The section processors 218 for a resume document type comprise a cover letter section processor 702, a contact information section processor 704, an experience section processor 706, an education section processor 708, an awards and honors section processor 710, a patents section processor 712, and a publications section processor 714. As described below in more detail, each section processor 218 analyzes a particular section in the electronic document 104 and extracts specific words and word groups from that section into a specific record in the target database 110. Additionally, as described below in more detail, each section processor 218 applies a set of heuristics to the particular section of interest in order to analyze and extract the desired information.

Cover Letter Section Processor

Some resumes may contain a cover letter at the beginning in which the applicant provides additional information. Typically, cover letters follow certain formatting and content conventions such as starting with a date and/or addresses of the sender and recipient, or a salutation. The cover letter usually ends with closings such as 'Sincerely' or 'Yours Truly' followed by the name of the sender. The xtraction server 108 stores information relating to these patterns and conventions in order to identify the beginning and the end of cover letters and registers the positions at which cover letter starts and ends in the document 104. Such information is used for identifying the boundaries of other sections such as the contact information section. The information extracted from the cover letter section includes information such as the beginning of the cover letter and the end of the cover letter.

The heuristics used for the cover letter section may comprise such assumptions or rules of thumb such as: a cover letter usually starts at the beginning of the document or immediately after e-mail headers if the document is a saved e-mail; a cover letter may contain a date, sender's and recipient's addresses, and a salutation (like 'Dear', 'To') at the beginning; and a cover letter ends with a complimentary closing (like 'Sincerely') and the name of the sender.

Contact Information Section Processor

As discussed above with reference to Figure 1, the xtraction server 108 follows a non-monotonic reasoning principle. Thus, the xtraction server 108 searches first for specific information then searches for other information. This principle is especially useful in extracting contact information. The contact information section generally includes an applicant's name, street address, city, state, zip code, phone numbers, and an e-mail addresss. The xtraction server 108 identifies where the contact information is present and extracts each of these pieces of information into a Candidate Record in the target database 110. Typically, information such as the zip code, state name, e-mail address, and phone numbers are definite and are provided in a fixed number of patterns. Thus, such information is preferably found first and the xtraction server 108 searches for other less definite information like city, street address and the name of the candidate.

Sometimes, an applicant may give two addresses (such as a work address and a home address) in a multi-column format. In a preferred embodiment, the xtraction server 108 separates these multi-column addresses as two blocks of contact information, determines which block is a work address and which block is a home address, processes each one of them separately, and stores the information in the corresponding database record in the target database 110. Separating the two address sections preferably involves two steps: checking if the contact information is a two-column, two-address type of a listing; and separating each column into a block of text. Specifically, the contact information section is processed to determine whether it has two addresses listed in two columns. Two column address listings may be identified by matching the address listing with various two-column patterns and by determining whether the spacing between the two columns is consistent and is formatted properly. After identifying the two-column address, the two addresses are placed in two separate buffers by calculating the boundaries of each column in a single row. Preferably, spacing and formatting data is used in this calculation.

Sometimes, the contact information section is listed in an attribute-value format. In other words, each piece of information presented is tagged by what that value is. For example, a resume may list a candidate's contact information as follows:

Name: Joe Smith
Address: 1234, Alta Vista Ave
Santa City, CA 98989
Phone: 465-989-7890 (Home); 465-989-0987 (Work)

In such a case, the values of different fields are obtained by looking at the header information without further processing. The xtraction server 108 maintains information headers and the related information. A table of attribute and value pairs is created using this knowledge and content of the contact information section. In a preferred embodiment, the information preferably extracted from the contact information section includes a first name, middle name, last name and salutation of the candidate, a street address, city, state, zip code, phone numbers, and an email address.

In a preferred embodiment, the contact information section processor first identifies the contact information section within document. The contact information section processor then determines whether a two column address block is present. If there is, then the processor separates the address blocks as described above. The contact information section processor then processes any tags that are present and determines whether certain information is present and extracts such information from the contact information section. For example, the contact information section processor searches for and extracts the zip code, state, email address, city, street address, and name of the applicant from the contact section of the document and stores the extracted information in a Candidate Record in the target database 110.

The contact information section processor preferably applies the following heuristics for the steps performed by the contact information section processor.

<u>Section identification:</u>

- The contact information usually appears either in the beginning or immediately after the cover letter. If a cover letter is identified within a resume, the end of the cover letter is assumed to be the beginning of the contact information section.
- The contact information section processor recognizes the end of the contact information section by looking at starting positions of various sections recognized by

the section processor. The lowest (or earliest) position within the document where a known section is starting is the end of contact information.

Processing tags, if present:

- The values of different fields are obtained by looking at the header information without further processing.
- The contact information section processor stores knowledge regarding specific types of headers and information.
- A table of attribute and value pairs is created using this knowledge and content of the contact information section.

Zip Codes:

- Zip codes of the United States follow fixed patterns such NNNNN or NNNNN-NNNN where N is a digit. Other countries have patterns, which may consist of fixed combinations of digits, alphabets and certain punctuation marks.
- These patterns are user configurable.
- With the knowledge of the user-configured patterns of zip codes, the contact information section processor examines each token and assigns the best-matched token as the value of the zip code.

State:

- The state names and their abbreviated forms are stored in the knowledge base of the xtraction server 108.
- This data is also configurable by the user.

Email:

- E-mail addresses follow a fixed pattern.
- E-mail addresses occur as unbroken tokens.

Phone Number:

- Phone numbers are strings of contiguous digits following a pattern.
- Most of the phone numbers listed also have an extra qualifier mentioning whether it is a home number or a work number.

City:

- After identifying the zip code, state, e-mail and phone numbers, it is easy to recognize the city by searching the area where the state and zip code were found.
- The piece of text before the state name and the zip code in the same line is recognized as the city.

- In case of failure, previous lines are considered for qualification. Further checks are done to ensure that the extracted string qualifies for a city name.

Street address:

- In the street address extraction the configurable knowledge of the conventions found in the names of streets is used. For example, the presence of keywords such as boulevard, street, or circle, indicates that the string is more likely to be a street address.
- If the potential string has digits at the beginning – it further qualifies for the street name.

Name:

- The name occurs above the rest of the information and it does not contain any numbers.
- Each token within the name starts with an uppercase alphabet.
- After qualifying a string to be the name, the name may be separated into the first name, last name, middle name and a salutation if present.
- The name of the candidate should not have any numbers.

Experience Section Processor

The experience section in a resume typically lists the experience of a candidate either in functional or in chronological order. Generally, the most common way of listing experience is in reverse chronological order - that is, listing the most recent job first and going backwards. Additionally, each job description commonly includes the employer name, duration, job title and the highlights of work done. Preferably, the experience section processor extracts the following information from the listing of each experience record: the start date of the job; the end date of the job; the job title; the employer name; and the highlights of the work done.

In a preferred embodiment, the experience section processor first identifies the experience section within the document, preprocesses the experience section, identifies the individual record listings, extracts the start and end dates from each of the listings, extracts job titles from each of the listings, extracts the employer name from each of the listings, extracts work highlights from each of the listings, performs post-processing steps on the extracted information, and stores the extracted information in an Experience Record in the target database 110.

In a preferred embodiment, the heuristics applied by the experience section processor includes:

Section Identification:

- The experience section starts at one of the positions that are identified as section beginnings based on the formatting information by the document pre-processor.
- The experience section starts with a keyword indicating the beginning of the experience section.
- The end of the experience section is a different position after the beginning of the experience section beginning and is already identified as a section beginning based on the formatting information by the document pre-processor.

Preprocessing:

- If the alphabets and the numbers occur together in the same string as in 'Jan98' or 'Win95', separate them as different strings as in 'Jan', '98' and 'Win', '95'.

Identify Individual Records:

- The pattern of experience listing, that is, if it is starting with a date or a job title, or employer name, repeats throughout the experience listing.
- Each listing is usually separated by the paragraph breaks.
- Each listing should have a minimum length.

Extracting Start and End Dates:

- Each date item is atomic, that is, all constituents of a single date occur together.
- Start and End dates occur together. but can optionally be separated by the prepositions like 'to' and/or new line breaks.
- If the end date is absent the year of start date must be of 'YYYY' format. That is, the date cannot be '98'. It should be '1998'.

Extracting Job Titles:

- All job titles specific to the particular industry are stored in the knowledge database. This data is fully configurable by the user.
- Job titles can be listed in any case, not necessarily in the case listed in the database.
- The string that is qualified as a job title cannot be considered for organization name.

Extraction Employer Name:

- Most employer names contain keywords like 'Corp.', 'Inc.', 'Company' etc. All these keywords are user configurable.

- Employer name is not split across the lines.
- In the absence of keywords, the presence of a state name or a date would qualify a line to be a potential employer name.

Extracting Highlights:

- Highlights if listed start with bullet characters like '*', '#', '-' etc.
- In the absence of bullet characters, don't extract highlights because the text of each experience listing is already stored.
- Important highlights are listed first

Post-Processing:

- If the experience listing starts with a date and the start and end dates are not extracted, disqualify the record and add the buffer to the buffer of previous record.
- If the employer name is not extracted for a listing, check the position where the employer name is recorded for most records and see if the string present in that position qualifies for the employer name.
- If no information on start date, end date, job title, employer name is extracted, disqualify that record and add the buffer to the buffer of previous record

Education Section Processor

In a resume document type, the education section typically lists the formal education and training obtained by a candidate. This is usually a short section listing higher and most recent degrees first. In this section, each listing may contain information such as the name of the degree, name of the major, date of completion, date started, university or the institute attended, GPA and the status as to whether the candidate has graduated or is still a candidate. In a preferred embodiment. the experience section processor extracts the following information from the listing of each education record: the start date of the degree, the end date of the degree, the degree name, the major name, the name of the university or the institute attended, the GPA, and the status of the degree.

In a preferred embodiment, the education section processor first identifies the education section within the document, identifies the individual record listings, extracts the start and end dates from each of the listings, extracts the degree name from each of the listings, extracts major names from each of the listings, extracts the university/institute name from each of the listings, extracts the GPA from each of the listings, extracts the status of the degree from each of the listing, and stores the extracted information in an Education Record in the target database 110.

In a preferred embodiment, the heuristics applied by the experience section processor includes the following:

Section Identification:

- The education section starts at one of the positions that are identified as section beginnings based on the formatting information by the document-preprocessing module.
- Education section starts with a keyword indicating the beginning of education section.
- The end of education section is a different position after the beginning of the education section beginning and is already identified as a section beginning based on the formatting information by the document-preprocessing module.

Identifying Individual Records:

- The pattern of education listing, that is, if it is starting with a date or a name of the degree, or university/institute name, repeats throughout the education listing.
- Each listing is usually separated by the paragraph breaks.
- Each listing should have a minimum length

Extracting Start and End Dates:

- Each date item is atomic, that is, all constituents of a single date occur together.
- Start and End dates occur together, but can optionally be separated by the prepositions like 'to' and/or new line breaks.
- If only one date is found. it should be treated as the date of completion.

Extracting Names of Degree and Major:

- All names of the degrees and majors are stored in the database. This data is fully configurable by the user.
- Degree and major names can be listed in any case, not necessarily in the case listed in the database.
- The string that is qualified as a degree or major cannot be considered for university or institute name.

Extracting University/Institute Name:

- The name of the university/institute contains keywords like 'University', 'Univ.', 'College' etc. All these keywords are user configurable.
- In the absence of keywords. the presence of a state name would qualify a line to be a potential university/institute name.

- The string containing the name of the university/institute may not contain punctuation marks other than a comma (',') and a period ('.').

Extracting GPA:

- GPA listing follows a pattern like N.NN or N.NN/NN.0 where N is a digit.
- GPA value usually follows the keyword 'GPA'

Extracting the Status of the Degree:

- The candidate mentions keywords like 'awarded', 'pending', 'expecting', 'candidate' etc. which give us the clue about the status of the degree.
- The status is usually mentioned next to the degree name or at the end of the listing

Awards and Honors Section Processor

In a resume type document, the awards and honors section typically lists the awards and honors received by the candidate. This is usually a short section listing items in one of the standard formats. In this section, each item is typically preceded by either a bullet character like '*', or listed in one paragraph, or as one line or in a multi-column format (several items in one line and the items are arranged in different columns). Each listing in this section may contain the date an award or an honor was obtained and a highlight of the award. In a preferred embodiment, the award and honors section processor extracts the following information from the listing of each awards and honors record: the date an award or honor was obtained, and the award or honor highlight.

In a preferred embodiment, the award and honors section processor first identifies the awards and honors section within the document, then recognizes the pattern or format of the listing, identifies the individual record listings, extracts the highlight of the award or honor, and stores the extracted information in an Award and Honor Record in the target database 110.

In a preferred embodiment, the heuristics applied by the awards and honors section processor includes:

Section Identification:

Heuristics similar to those discussed above with reference to the experience and education section processors are used by the awards and honors section processor for the section identification process.

Recognizing the Pattern of Listing and Identifying Individual Records:

- If the section starts with a bullet character and there are other lines in the buffer starting with bullet characters then the listing is of bulleted format.

- If the section can be divided into paragraphs the pattern is of paragraph format.
- If the each line within the section has multiple columns, the listing is of multi-column format.
- Use the knowledge of the formatting to identify individual records.

Extracting Highlights:

- The content of the listed item becomes the highlight.

Patents Section Processor

In some resume document types, there may be a patents section which lists the patents obtained or filed by a candidate. This section lists items in one of the standard formats. In this section, each item is typically preceded by either a bullet character like '*', or listed in one paragraph, or as one line or in a multi-column format (several items in one line and the items are arranged in different columns). Each listing in this section may contain the date when the patent was granted or filed, title of the patent, patent number, and the status of the patent (whether the patent is filed or granted). In a preferred embodiment, the patent section processor extracts the following information from the listing of each patent record: the date when the patent was filed/granted, the name of the patent, the patent number, and the status of the patent.

In a preferred embodiment, the patents section processor first identifies the patents section within the document. Then the patents section processor recognizes the pattern or format of the listing, identifies the individual record listings, extracts the date when a patent was granted/filed, extracts the title of the patent, extracts the status of the patent, extracts the patent number, and finally, stores the extracted information in a Patents Record in the target database 110.

In a preferred embodiment, the heuristics applied by the patents section processor include:

Section Identification:

Heuristics similar to those that were described with reference to the experience and education section processor are used.

Recognizing the Pattern of Listing and Identifying Individual Records:

Heuristics similar to those described with reference to the awards and honors section processor are used.

Extraction Patent Number:

- Patent number patterns are stored in the knowledge base of the extraction server, which is user configurable.
- Examine individual strings of the patents section if they match with the known patterns.

Extracting Date:

- Date occurs atomically

Status of the Patent:

- Look for the keywords like 'Pending', 'granted' etc. The presence of these keywords determines the status of the patent.
- In absence of such keywords, assume the status of patent as 'granted'.
- If the patent number is found, the status of the patent is 'granted'.

Extracting Title of the Patent:

- The content of the listed item other than the date, patent number, patent status is recorded as the name of the patent.

Publications Section Processor

In some resume document types, the resume may contain a publications section which lists the books, technical articles, journal articles and any other publications by the candidate. This section typically lists items in one of the standard formats. In this section, each item is either preceded by a bullet character like '*', or listed in one paragraph, or as one line or in a multicolumn format (several items in one line and the items are arranged in different columns). Each listing in this section may contain the date of publication, publication name, publisher name, publication type, ISBN number if any, page range if any. In a preferred embodiment, the publications section processor extracts the following information from the listing of each publication record: the date of publication, the ISBN, the page range, the publication type, the publication name, and the publisher name

In a preferred embodiment, the publications section processor first identifies the publications section within the document. The publications section processor then recognizes the pattern or format of the listing, identifies the individual record listings, extracts the date of the publication, extracts the ISBN, extracts the page range, extracts the publication type, extracts the publication name, extracts the publisher name, and finally, stores the extracted information in a Publications Record in the target database 110.

In a preferred embodiment, the heuristics used by the publications section processor include:

Section Identification:

Heuristics similar to those that were described with reference to the experience and education section processors are used.

Recognizing the Pattern of Listing and Identifying Individual Records:

Heuristics similar to those described with reference to the awards and honors section processor are used.

Extracting ISBN:

- ISBN patterns are stored in the knowledge base of the extraction server, which is user configurable.
- Examine individual tokens of the 'Publications' section if they match with the known patterns.

Extracting Page Range:

- Page range is usually followed by the keywords such as 'pp.' or 'pages' etc.
- Page range listings follow the pattern such as N-N, where N is a number.

Extracting Date:

- Date occurs atomically

Publication Type:

- Look for the keywords like 'technical report', 'journal', 'article' etc. The presence of these keywords determines the status of the patent.
- If the publications are listed under sub headings like 'Books', 'Technical Articles' etc. all items listed under these subheadings will have the type mentioned above the listing.

Extracting Publication Title:

- If there is some content with in the quotation marks, that will be taken as the publication name.
- In the absence of above condition the content other than the type, status and date, page range is taken as the publication title.

Extracting Publisher Name:

- If the publication title is found with in the quotation marks then the content other than the type, status and date, page range and publication title is taken as the publisher name.
- Publisher name usually consists of keywords like 'Publishing Co.', 'Publishers' etc. These keywords are also user configurable.

The Target Database

As described above with reference to Figure 1, the target database 110 comprises predefined records or database tables with predefined columns for storing the word and word groups which are extracted from the electronic document 104. The target database 110 may comprises several records for storing information for a particular document type. In a preferred embodiment, the target database 110 for a resume document type comprises a Candidate Record for storing personal information about the candidate which has been extracted from the resume, an Experience Detail Record for storing information about the candidate's work experience, an Education Record for storing information pertaining to the degree or education mentioned in the resume. an Award-Honors Record for storing information pertaining to awards or honors received by the candidate, a Course Record for storing information pertaining to courses taken or diplomas received by the candidate. a Patent Record for storing information pertaining to patents for which the candidate is an inventor, and a Publication Record for storing information pertaining to any publications about the candidate. One skilled in the art with realize that the target database 110 is not limited to the records, columns headings, and descriptions illustrated in the following tables, but that the target database may include any type of record, table format, column headings, and descriptions based on the document type.

Table 9 illustrates the preferred column headings and descriptions for a Candidate Record for storing personal information about a candidate.

Table 9

CandidateID	Candidate ID. the database ID of the person
CAFirstName	First Name of the Person
CALastName	Last Name of the Person
CANickName	Nick Name of the Person
CASalutation	Salutation (Mr.. Ms. Dr etc)

CAWorkCompany	Current Employer
CATitle	Current Designation
CAYearsEmployed	Total Number of years of experience
CACurrentObjective	Stated Objective in the Resume
CABriefExperience	Text of the Summary Section
CAYearsOfExperienc	Years of Experience with current employer
CAWorkStreet	Street Name of the Work address
CAWorkSuiteNo	Suite Number of Work address
CAWorkCity	City name of the Work Address
CAWorkState	State Name of the Work Address
CAWorkZip	Zip Code of the Work Address
CAWorkCountry	Country of the Work Address
CAWorkMailStop	Mail Stop of the Work Address
CAWorkPhoneNo	Work Phone Number
CAWorkExtension	Work Phone Number Extension
CAWorkFaxNo	Work Fax Number
CAWorkMobilePhone	Work Related Mobile Phone Number
CAWorkEmail	Official Email Address
CAHomeStreet	Street Name of the Home Address
CAHomeSuiteNo	Suite/Apt. Number of the Home
CAHomeCity	City of Residence Address
CAHomeState	State of Residence Address
CAHomeZip	Zip code of the Residence Address
CAHomeCountry	Country of the Residence Address
CAHomeMailStop	Mail Stop of Residence
CAHomePhoneNo	Home Phone Number
CAHomeExtension	Home Phone Extension
CAHomeFaxNo	Home Fax Number
CAHomeMobilePhone	Mobile phone of other address
CAHomeEmail	Home Email Address
CAHomePage	Web address
CAPager	Pager Number

CAOtherStreet	Street Name of address other than work and home
CAOtherSuiteNo	Suite or Apt. Number other than work and phone
CAOtherCity	Name of City from the other address
CAOtherState	State Name of the other address
CAOtherZip	Zip Code of the other address
CAOtherCountry	Country of the other address
CAOtherMailStop	Mail Stop other than Office and Residence
CAOtherPhoneNO	Any other Phone Number (e.g. Recruiting Agency)
CAOtherExtension	Extension number of Phone
CAOtherFaxNo	Fax Number other than work and residence
CAOtherMobilePhone	Mobile Phone Number
CAOtherEmail	Email Address other than residence and work
CALastModifiedDate	Date of Last Modification to the Record
CATextResume	Actual text of the resume
CAModifiedBy	Person who modified the record

Table 10 illustrates the preferred column headings and descriptions for an Experience Detail Record for storing information pertaining to a candidate's experience in the target database 110. Preferably, the database record illustrated in Table 10 stores the information extracted from the resume pertaining to one project or a job done at a particular company. A candidate may have more than one Experience Detail Record. An Experience Detail Record is created for each of the projects that were mentioned in the experience section of the resume.

Table 10

CandidateID	Database ID of the Person (Candidate Table)
EDEmployerName	Name of the Company worked for
EDReportedTo	Name of the reporting Manager
EDResponsibilityL1	Primary Responsibility (Designation)
EDResponsibilityL2	Secondary Responsibility
EDPeopleManaged	Number of people managed
EDHighlights1	First Bulleted item from the Experience Description
EDHighlights2	Second Bulleted Item

EDHighlights3	Third Bulleted Item
EDHighlights4	Fourth Bulleted Item
EDNotes	Text of the Experience Description
ExperienceDetailID	ID of the Record
EDStartDateDD	Date joined for the Company
EDStartDateMM	Month joined for the Company
EDStartDateYYYY	Year joined for the company
EDEndDateDD	Date last worked for the company
EDEndDateMM	Month last worked for the company
EDEndDateYYYY	Year last worked for the company
EDReportedToPhone	Manager's Phone Number

Table 11 illustrates the preferred column headings and descriptions for an Education Record. This record stores the information pertaining to the degree or education that was mentioned in the resume. Preferably, a single record is created for each degree mentioned on the resume.

Table 11

CandidateID	Database ID of the person
EdrfDegreeType	Type of the Degree (BS. MS, PhD)
EdrfMajor	Specialization
EDAwardedby	Name of the Institution
EDGPA	GPA earned
EdrfGradStatus	Status of graduation (passed, pending, etc...)
EDStartDateDD	Date joined in the course
EDStartDateMM	Month of joining
EDStartDateYYYY	Year of joining
EDEndDateDD	Date of completion
EDEndDateMM	Month of completion
EDEndDateYYYY	Year of completion
EDNote	Text of the description of the record

Table 12 illustrates the preferred column headings and descriptions for an Awards-Honors Record.

Table 12

AWhighlight	Name and highlight of the Award or Honor
CandidateID	Database ID of the candidate
AWNotes	Description of the Award or Honor

Table 13 illustrates the preferred column headings and descriptions for a Course Record.

Table 13

COCourseName	Name of the Course taken
CandidateID	Database Id of the Candidate
CODDateDD	Date course taken (date)
CODateMM	Date course taken (month)
CODateYYYY	Date course taken (year)
CONotes	Description of the course

Table 14 illustrates the preferred column headings and descriptions for a Patent Record. Preferably, a single record is created for each of the patent mentioned in the resume.

Table 14

CandidateID	Database Id of the Person
PATitle	Title of the Patent
Pacountry	Country where Patent was filed
PAJointHolder	Name of the Joint Holder
PAPatentNumber	Patent Number
PAGrantDateYYYY	Year Patent Granted
PAPatentStatus	Status of the Patent (granted, pending)
PANotes	Text of the description of the Patent
PAGrantDateMM	Month Patent granted
PAGrantDateDD	Date Patent granted

Table 15 illustrates the preferred column headings and descriptions for a Publication Record. Preferably, a single record is created for each publication mentioned by the candidate.

Table 15

CandidateID	Database Id of the Person
PurfPublicatType	Type of Publication (Book, Paper, etc...)
PUTitle	Title of Publication
PUPublicationName	Name of the Publication
PUDateDD	Date of Publication
PUPublisherName	Name of the Publisher
PUDateMM	Month of Publication
PUPageRange	Page Numbers
PUDateYYYY	Year of Publication
PUisbn	ISBN number of the Publication
PUNotes	Text of the description

The above description is included to illustrate the operation of the preferred embodiments and is not meant to limit the scope of the invention. The scope of the invention is to be limited only by the following claims. From the above discussion, many variations will be apparent to one skilled in the art that would yet be encompassed by the spirit and scope of the present invention.

What is claimed is:

Claims

1. A system for extracting words and word groups from an electronic document and for storing extracted words and word groups into a target database, the system comprising:

a content analysis and semantic network engine for analyzing and extracting words and word groups from the electronic document; and

a heuristics engine coupled to the content analysis and semantic network, for applying a set of heuristics to the words and word groups in the electronic document.

2. The system of claim 1 further comprising:

a document pre-processor coupled to the content analysis and semantic network engine, for performing an initial analysis on the electronic document.

3. The system of claim 1 wherein the content analysis and semantic network engine further comprises:

a thesaurus for linking together terms and concepts; and

a semantic network, linked to the thesaurus, for organizing terms and concepts of the thesaurus, and meta-concepts, and categories, and for defining relationships between and among the terms, concepts, meta-concepts, and categories.

4. The system of claim 3 wherein the semantic network is based on subsumption.

5. The system of claim 1 wherein the electronic document comprises a document type having a plurality of sections, and a set of heuristics is applied to each section of the electronic document.

6. The system of claim 1 further comprising:

a morphological analysis engine coupled to the heuristics engine for performing a morphological analysis and tagging of words and word groups in the electronic document.

7. A system for analyzing and extracting words and word groups from an electronic document into a target database having predefined fields, the apparatus comprising:

a thesaurus for linking together terms and concepts and for defining relationships between and among terms and concepts; and

a semantic network coupled to the thesaurus, for organizing terms and concepts in the thesaurus, meta-concepts, and categories in a hierarchical structure;

wherein the thesaurus and semantic network are used to analyze words and word groups in the electronic document.

8. The system of claim 7 further comprising:

a document pre-processor coupled to the semantic network, for classifying the document as a document type and for performing an initial analysis on the electronic document.

9. The system of claim 7 further comprising:

a heuristics engine coupled to the semantic network, for applying a set of heuristics to the electronic document.

10. The system of claim 7 further comprising:

a target database coupled to the semantic network, for storing the words and word groups from the electronic document in the predefined fields in the target database.

11. The system of claim 7 wherein the electronic document comprises a plurality of sections and the system further comprises:

section processors for analyzing a section in the document and applying a set of heuristics to each section in the document.

12. The system of claim 7 further comprising a database interface for interfacing with the target database, said database interface coupled to the semantic network.

13. A method for extracting words and word groups from an electronic document, the method comprising the steps of:

identifying a section in the electronic document;
analyzing the section in the electronic document;
extracting words and word groups from the section in the electronic document; and
storing words and word groups extracted from the section into a target database.

14. The method of claim 13 wherein the step of identifying a section in the electronic document is performed by applying a set of identification heuristics to the section.

15. The method of claim 13 wherein the step of analyzing the section in the electronic document is performed using a semantic network.

16. The method of claim 15 further comprising the step of:

analyzing the relationships between and among words and word groups in the electronic document using the semantic network.

17. The method of claim 13 further comprising the steps of:

converting the electronic document from a native file format into an ASCII text format;

filtering out unnecessary information from the electronic document; and

identifying sections in the electronic document.

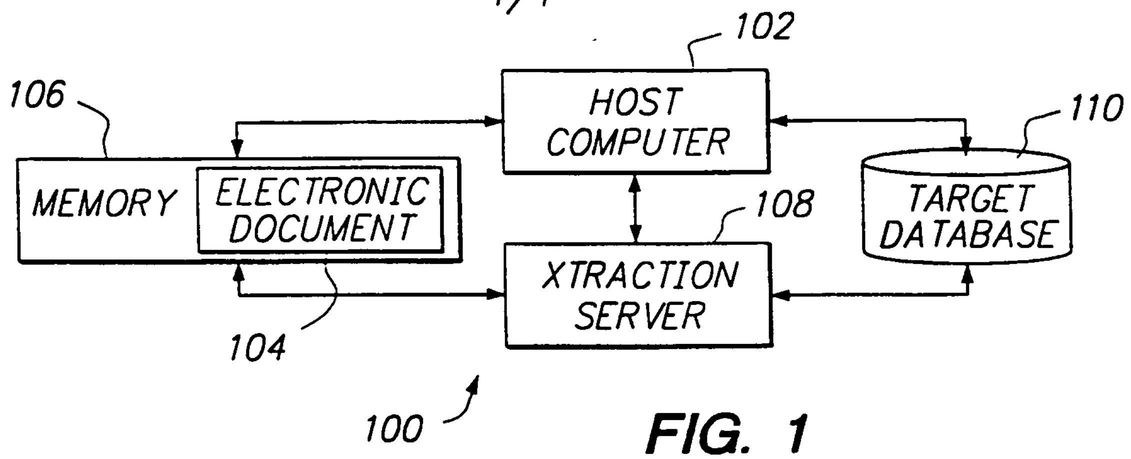

102
HOST COMPUTER
106
MEMORY
ELECTRONIC DOCUMENT
110
TARGET DATABASE
108
XTRACTION SERVER
104
100

FIG. 1

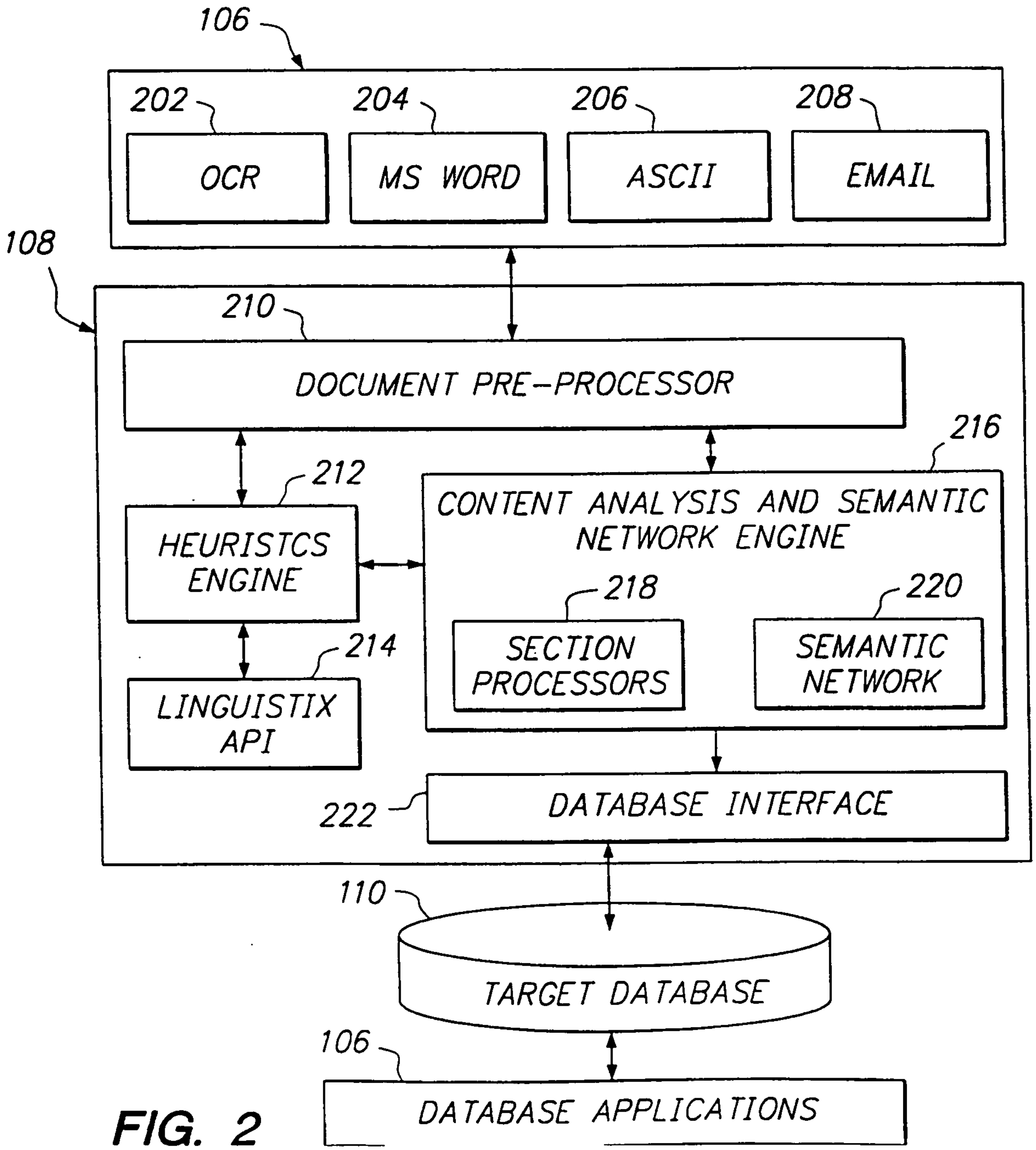

106
202
OCR
204
MS WORD
206
ASCII
208
EMAIL
108
210
DOCUMENT PRE-PROCESSOR
216
212
HEURISTCS ENGINE
CONTENT ANALYSIS AND SEMANTIC NETWORK ENGINE
218
SECTION PROCESSORS
220
SEMANTIC NETWORK
214
LINGUISTIX API
222
DATABASE INTERFACE
110
TARGET DATABASE
106
DATABASE APPLICATIONS

FIG. 2

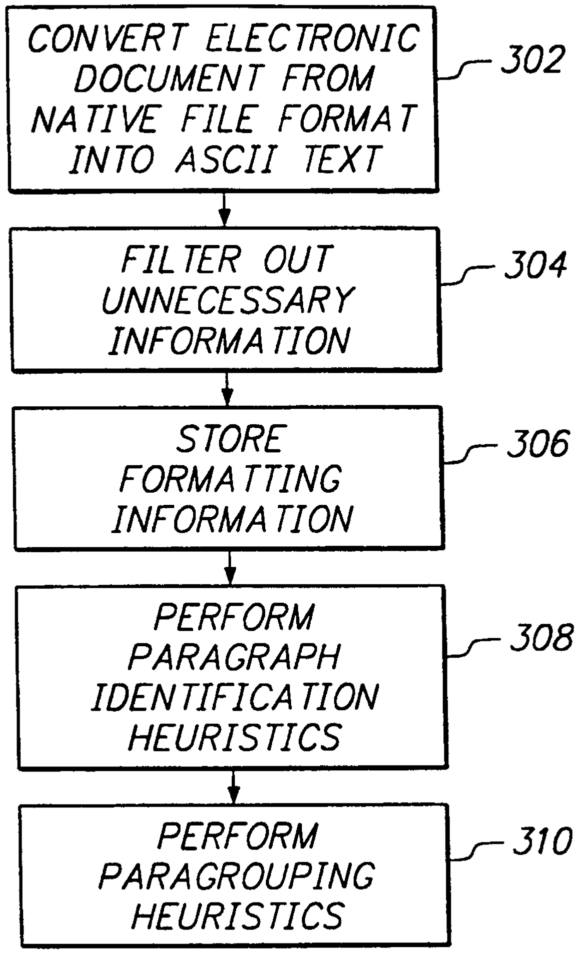

CONVERT ELECTRONIC DOCUMENT FROM NATIVE FILE FORMAT INTO ASCII TEXT
302
FILTER OUT UNNECESSARY INFORMATION
304
STORE FORMATTING INFORMATION
306
PERFORM PARAGRAPH IDENTIFICATION HEURISTICS
308
PERFORM PARAGROUPING HEURISTICS
310

FIG. 3

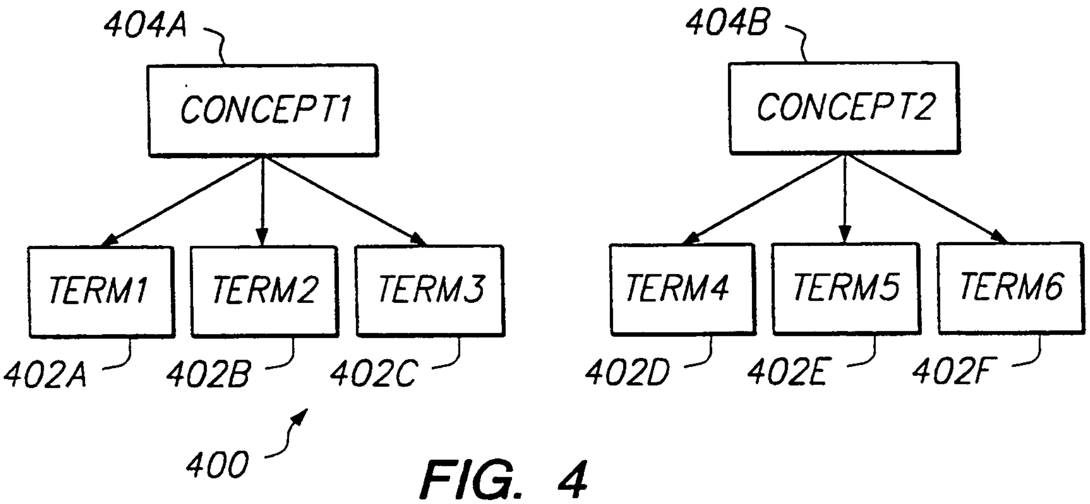

404A
CONCEPT1
404B
CONCEPT2
TERM1
TERM2
TERM3
TERM4
TERM5
TERM6
402A
402B
402C
402D
402E
402F
400

FIG. 4

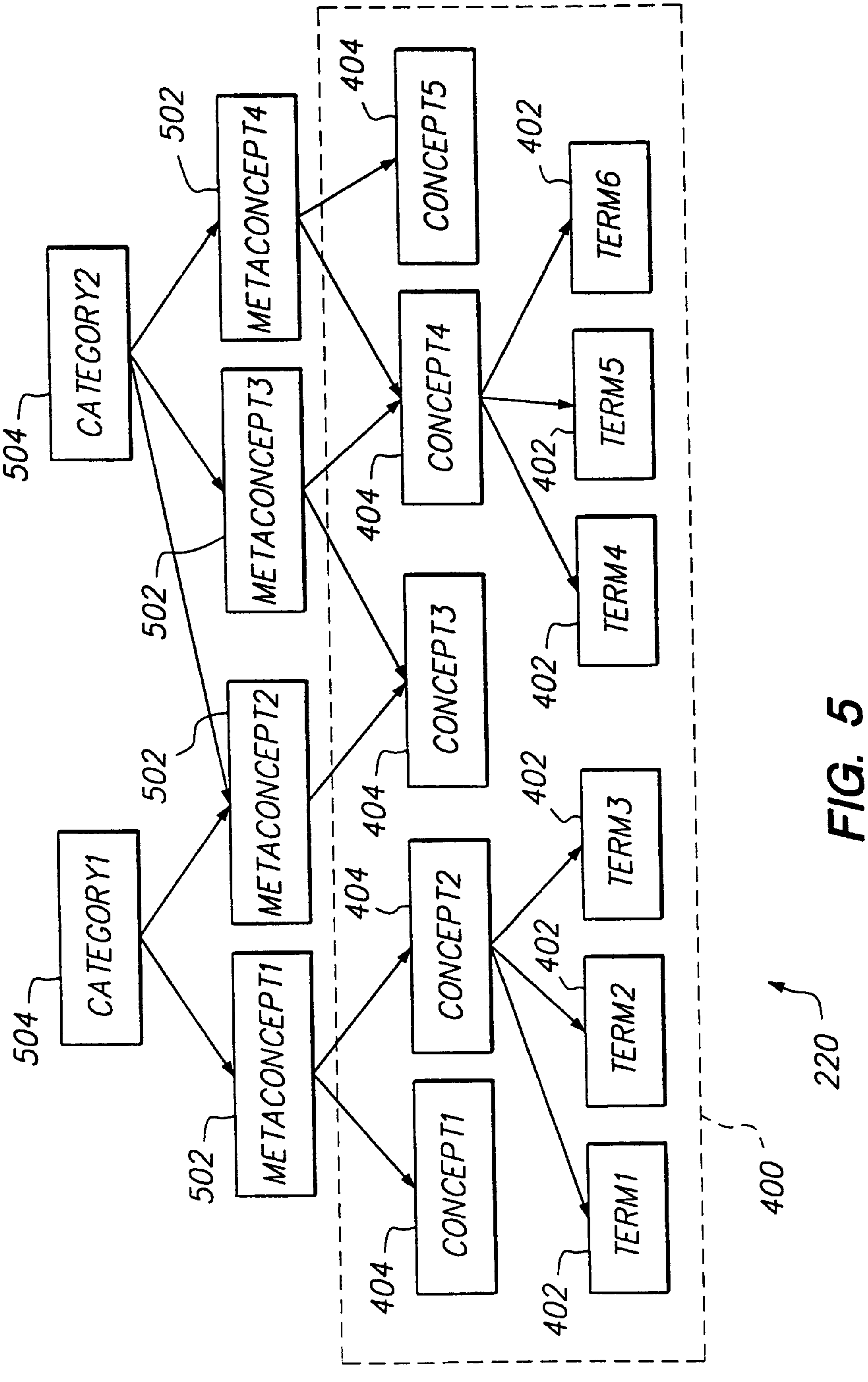

CATEGORY1
CATEGORY2
504
504
METACONCEPT1
METACONCEPT2
METACONCEPT3
METACONCEPT4
502
502
502
502
CONCEPT1
CONCEPT2
CONCEPT3
CONCEPT4
CONCEPT5
404
404
404
404
404
TERM1
TERM2
TERM3
TERM4
TERM5
TERM6
402
402
402
402
402
402
400
220

FIG. 5

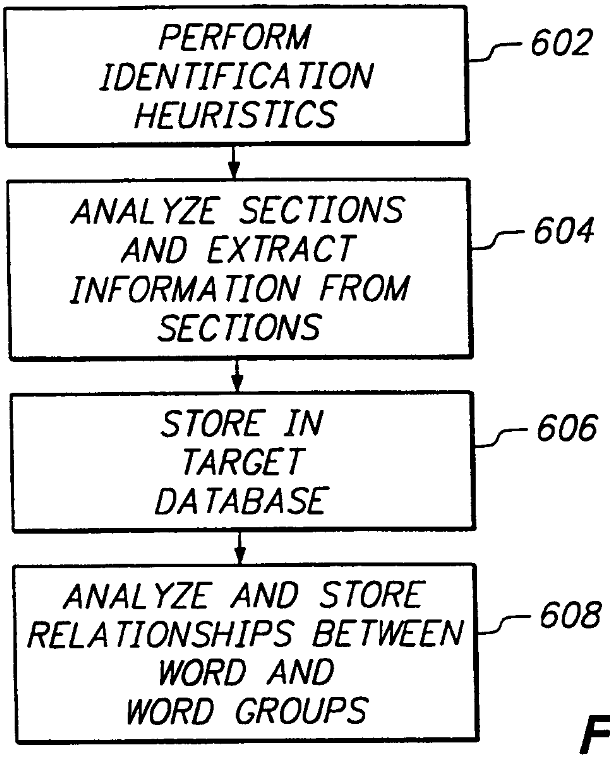

PERFORM IDENTIFICATION HEURISTICS
602
ANALYZE SECTIONS AND EXTRACT INFORMATION FROM SECTIONS
604
STORE IN TARGET DATABASE
606
ANALYZE AND STORE RELATIONSHIPS BETWEEN WORD AND WORD GROUPS
608

FIG. 6

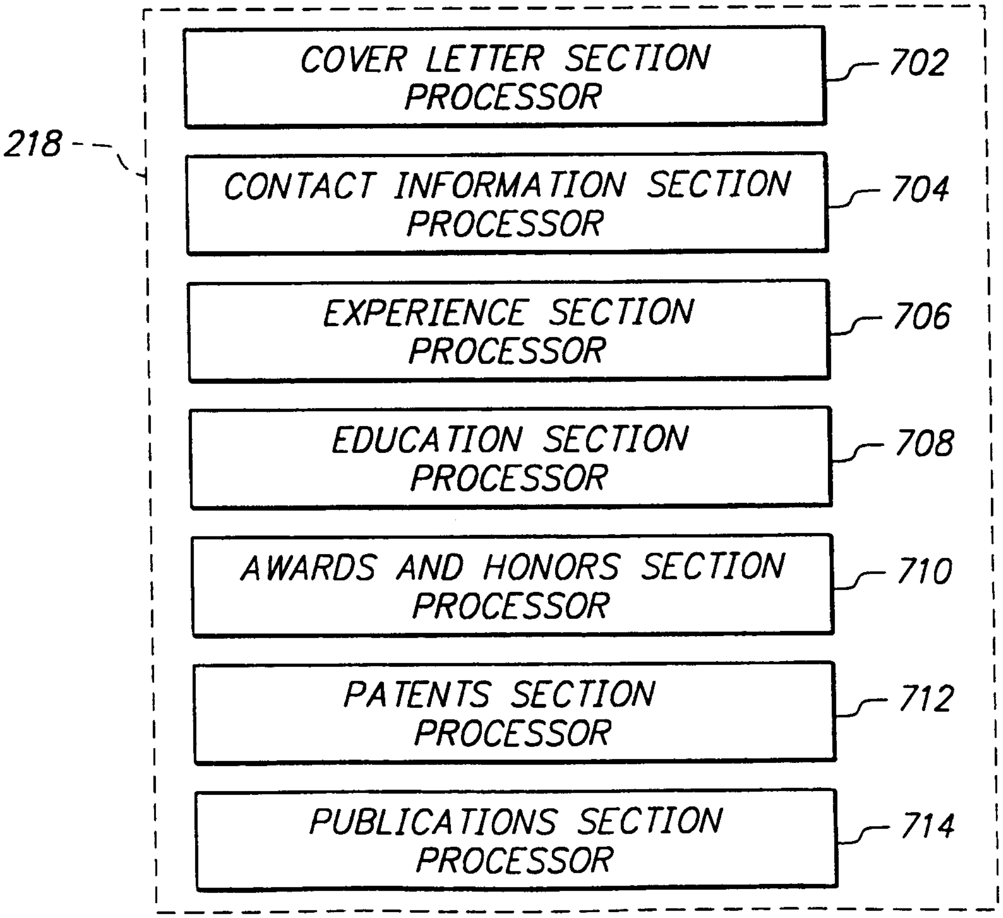

218
COVER LETTER SECTION PROCESSOR
702
CONTACT INFORMATION SECTION PROCESSOR
704
EXPERIENCE SECTION PROCESSOR
706
EDUCATION SECTION PROCESSOR
708
AWARDS AND HONORS SECTION PROCESSOR
710
PATENTS SECTION PROCESSOR
712
PUBLICATIONS SECTION PROCESSOR
714

FIG. 7

INTERNATIONAL SEARCH REPORT

Int .tional Application No
PCT/US 98/27664

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	HAMMER J ET AL: "Extracting semistructured information from the Web" PROCEEDINGS OF THE WORKSHOP ON MANAGEMENT OF SEMI-STRUCTURED DATA, PROCEEDINGS OF WORKSHOP ON MANAGEMENT OF SEMI-STRUCTURED DATA, TUCSON, AZ, USA, 16 MAY 1997, pages 18-25, XP002099172 1997, Murray Hill, NJ, USA, AT & T Labs - Research, USA see the whole document	1,2,13, 14
A		3-12,15, 16

	-/--	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search 8 April 1999	Date of mailing of the international search report 22/04/1999
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Abbing, R

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	ASHISH N ET AL: "Semi-automatic wrapper generation for Internet information sources" PROCEEDINGS OF THE SECOND IFCIS INTERNATIONAL CONFERENCE ON COOPERATIVE INFORMATION SYSTEMS, COOPIS'97 (CAT. NO.97TB100143), PROCEEDINGS OF COOPIS 97: 2ND IFCIS CONFERENCE ON COOPERATIVE INFORMATION SYSTEMS, KIAWAH ISLAND, SC, USA, 24-27 JUNE 1997, pages 160-169, XP002099173 ISBN 0-8186-7946-8, 1997, Los Alamitos, CA, USA, IEEE Comput. Soc, USA see page 163, column 1, line 16 - page 166, column 1, line 19 ---	1-16
Y	SMITH D ET AL: "Information extraction for semi-structured documents" PROCEEDINGS OF THE WORKSHOP ON MANAGEMENT OF SEMI-STRUCTURED DATA, PROCEEDINGS OF WORKSHOP ON MANAGEMENT OF SEMI-STRUCTURED DATA, TUCSON, AZ, USA, 16 MAY 1997, pages 60-66, XP002099174 1997, Murray Hill, NJ, USA, AT & T Labs - Research, USA see the whole document ---	1-16
A	NESTOROV S ET AL: "Inferring structure in semistructured data" SEMI-STRUCTURED DATA WORKSHOP HELD IN CONJUNCTION WITH SIGMOD '97, TUCSON, AZ, USA, MAY 1997, vol. 26, no. 4, pages 39-43, XP002099175 ISSN 0163-5808, SIGMOD Record, Dec. 1997, ACM, USA see the whole document ---	1-16
A	US 5 297 039 A (KANAEGAMI ATSUSHI ET AL) 22 March 1994 see abstract see column 2, line 7 - column 6, line 59 -----	1,3,7,13

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte .ional Application No
PCT/US 98/27664

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5297039 A	22-03-1994	JP 4357568 A	10-12-1992